



Telerobotics over ip networks: Towards a low-level real-time architecture

Arnaud Lelevé, Philippe Fraisse, Pierre Dauchez

► To cite this version:

Arnaud Lelevé, Philippe Fraisse, Pierre Dauchez. Telerobotics over ip networks: Towards a low-level real-time architecture. 2001 IEEE IROS, Oct 2001, Maui, Hawaii, United States. pp.643-648, 10.1109/IROS.2001.976242 . hal-00189072

HAL Id: hal-00189072

<https://hal.science/hal-00189072>

Submitted on 16 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Telerobotics over IP Networks: Towards a Low-level Real-time Architecture

A. LELEVÉ, P. FRAISSE, P. DAUCHEZ

LIRMM - UMR 5506 CNRS / Université Montpellier II
161 rue Ada - 34392 Montpellier CEDEX 5 - France

Long distance teleoperation over asynchronous transmission links makes many classical teleoperation schemes unstable. The use of this kind of media involves varying transmission delays that may become prohibitive even at short distance. However, they are standardized, cheap and widespread over the planet. This paper presents our latest works on the improvement of teleoperation loops, by providing a low-level architecture which permits the use of classical teleoperation controls over asynchronous digital links.

1. PRELIMINARIES AND PROBLEM DESCRIPTION

1.1 Introduction

There are situations when firms or laboratories have to resort to remote manipulation. Such cases appear when dangerous objects have to be handled [1] or/and when the environment is too aggressive for humans. Typical applications belong to the nuclear domain (for instance in the dismantling of a nuclear plant), deep-sea domain (work on underwater structures of oil rigs) and spatial domain (exploration of distant planets).

Teleoperation has the supplementary advantage of giving the possibility of sharing an experiment between several operators located in distinct places. This way, heavy outdoor experimentations could be easily shared between several laboratories and costs could be reduced as much.

However, long distance control of a remote system requires the use of different transmission media which causes two main technical problems in teleoperation: limited bandwidth and transmission delays due to the propagation, packetisation and many other events digital links may inflict on data [2]. Moreover bandwidth and delays may vary according to events occurring all along the transmission lines. In acoustic transmission, round-trip delays greater than 10s and bit-rates smaller than 10kbits/s are common.

These technical constraints result in one hand in difficulties for the operator to securely control the remote system and, in the other hand, make classical controls unstable. Many researches have proposed solutions when delays are small or constant (for instance [3]), but when delays go beyond a few seconds and vary a lot as over long distances asynchronous links, solutions not based on teleprogramming [4] are fewer because such delays make

master and slave asynchronous and the control unstable.

This paper gives details on low-level teleoperation technique we have adopted when transmissions go over an asynchronous computer networks such as local IP networks and the *Internet* when necessary. Our structure features constant delays and synchronous data transmission. It also permits to easily implement a prediction/estimation function as soon as one has a realistic model of the teleoperated system. As soon as this structure gets into steady pattern, one can use any high-level teleoperation controls such as in [3].

2. OUR TELEOPERATION ENVIRONMENT

2.1 Strategy

For a few years, we have worked on improving teleoperation architecture through asynchronous networks. We have started this project by making experimentations [5] with a minimal architecture which helped us to model a generic teleoperation loop. We have used this model in simulation to test several solutions. We have oriented our researches on a low level architecture likely to be used by and to improve performances of any high level teleoperation method such as shared or bilateral control.

In this paper, we use an *IP* network over an *Ethernet LAN* with possibilities to be extended to the *Internet*.

Next paragraphs introduce the generic teleoperation architecture we base our study on and the teleoperation platform we have built.

Section 3 reminds the model we have obtained from our first experimentations and section 4 details the two main points we have recently focused on: network delays regulation and remote state prediction/estimation technique. This section also presents latest simulation results we have obtained.

2.2 Teleoperation Architecture

We consider a classical teleoperation diagram including two parts (the master alias *Base* and the slave alias *Remote System*) separated by a network transmission link. Figure 1 gives a global view of the teleoperation diagram and details what each part includes.

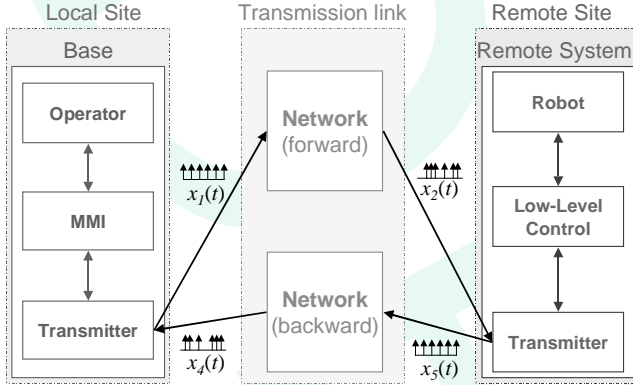


Figure 1 - Teleoperation Architecture

We will use the term "*Base*" to refer to the host computer that features the operator and the man-machine interface (MMI). The "*Remote System*" will point out the system to be teleoperated, i.e. the transmitter and the low-level control associated with the robot.

To simplify the reasoning, we will consider a monodimensional system. We just make the assumption that every axis of a complex system can be independently controlled, which is feasible by mean of computed torque methods, for instance.

Signals exchanged between the master and the slave are periodically emitted back and forth every 200 ms before being delayed by the network both way. This period was the smallest our platform could handle in the first experiments.

2.3 Teleoperation Platform

We use a terrestrial vehicle equipped with a 6 degree-of-freedom manipulator (d.o.f.) [5]. It consists of a 6x6 vehicle fitted with a *PUMA* manipulator. Several control laws are featured: global movement of the whole mobile manipulator [6], force-driven control laws, *PID* control law, ... In this present work, we use a computed torque method [7] to control the arm. A *dSpace*® DSP board is dedicated to the low-level control of the 8 axis (6 for the arm and 2 for the vehicle). A laptop PC fitted with a wireless *2Mbps* Ethernet network board takes care of the transmission and of a global control of the whole mobile manipulator.

2.4 Global Communication Architecture

The operator sends and receives data to the remote site through his lab network. This data roams through the Internet if necessary, it reaches the remote site LAN and,

at last, it is sent at 2Mbps through a 802.11-like radio transmission link. Figure 2 pictures this trip.

We have chosen *TCP* versus *UDP* because *TCP* is liable: the received data is exactly the same as the sent data (no packet loss or disorder). Details of the socket implementation has been published in [9].

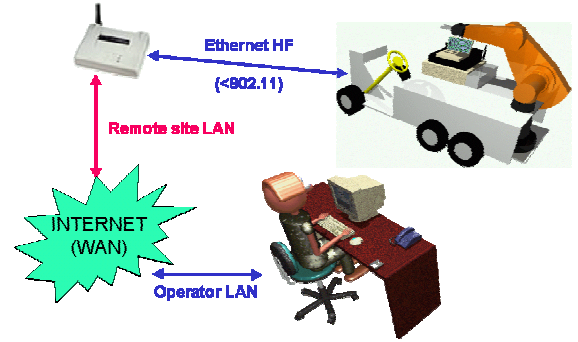


Figure 2 – Transmission loop

3. MODELISATION

3.1 Initial Experimentations

In order to build a mathematical model for our teleoperation tasks, we have performed preliminary experiments (depicted in [5]) using the experimentation platform introduced in section 2.3.

These various experiments have led in one hand to a model of the transmission link as a *Poisson*-like random behavior which μ parameter evolves according to the distance between local and remote sites. In the other hand, *Remote System* has been modeled as a monodimensional 2nd order discrete low-pass filter as described in [9].

These results allowed us to build a generic monodimensional model for simulation purposes under MATLAB/SIMULINK® environment [9].

3.2 Effects of network delays on signals

Globally, the effects of both forward and backward network delays are visible in figure 3 particularly on $x_3(t)$: transmission lines don't only delay the signals but they also distort them by making their sampling period vary. This distortion makes the signals, in one hand, not representative of the operator desired movements, and in the other hand, unusable for any remote analysis of the state of the remote site. This is why we have worked at first on a solution to get an accurate image of original data after a trip through an asynchronous transmission link.

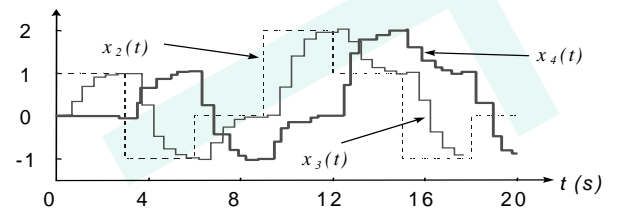


Figure 3- Effects of the network on $x_3(t)$

4. STUDIES

4.1 Introduction

This section will introduce you the low-level transmission architecture we have set up. It features a mean to easily compute an estimation of the state of the remote system (before backward data has reached the *base*) and also a prediction of movements at the moment when corresponding data is generated by the operator. This assumes delays have been made constant, which is the first subject we will develop.

4.2 Network Delay Regulation

4.2.1 Principle

In order to eliminate the delays jitter that is typical in asynchronous networks, we have thought of inserting two "Network Delay Regulator" (*N.D.R.*), located at the input of both *Base* and *Remote System*. Each one stacks incoming data as it is asynchronously received; meanwhile each *NDR* unstacks previous data (in the same order : *First-In, First-Out*) at a constant sampling rate (T_i) common to the whole teleoperation structure. The results are that signals find again the initial shape they initially had. The main drawback of this operation is the growth of the mean delay. In our study, we have observed the network round-trip-time (*NRTT*) and the global round-trip-time (*GRTT*) including *NRTT* and both *NDRs* through time. Figure 4 illustrates the difference between *NRTT* and *GRTT*. Figure 5 illustrates the effects of the *NDR* on delays with and without regulation.

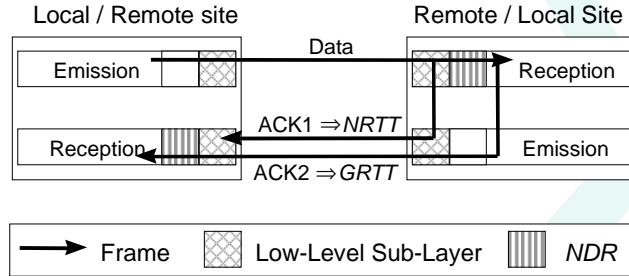


Figure 4 - NRTT and GRTT

In a first approach, we have assumed that both sites are too far to be synchronized so that we don't have access to single-trip-times. We have had to assume that forward and backward trip times can be deduced from *NRTT* by dividing it by 2. We are currently working on this subject; several solutions are conceivable: we could try to synchronize both hosts through the network using methods like [11] or using GPS clock up to a resolution of 500ns, which is quite enough for our experimentations.

4.2.2 Queue dimensioning

In order to fit the size of the *NDRs* queue to the network behavior, there is an initialization pattern at the beginning of teleoperation experiment. 10 frames are exchanged in

order to measure the network round-trip delays.

These measures are then analyzed; we use the maximum time (mt) between two receptions (ie maximum difference between delays) of these *NRTT* to set the nominal size (ns) of the *DVC* queue, with a factor of security η :

$$ns = 2 + E \left(\eta \times \frac{mt}{T_i} \right)$$

When $\eta=1$, the queues can handle delays corresponding to a maximum time between two receptions equal to mt . Whenever this case occurs, the queue empties down to 2 samples, the minimum size we permit.

$\eta=1.3$ ensures that the queue won't go beyond this limit even if network delays deteriorate a bit. Moreover, this security factor limits the lack of precision of the *NRTT* measure, due to the initialization pattern short duration.

4.2.3 Experimental Results

We have experimented the *NDR* at short distance (two hosts 75km apart) and at virtual long distance (a relay delayed network data with values corresponding to a transmission between Paris, France and New York, USA).

Figure 5 allows to compare the *Network Round-Trip-Time* (*NRTT*) with the *Global Round-Trip-Time* (*GRTT*) in steady pattern. At long distance, statistics give a mean *GRTT* to 7s (vs. 610ms for the *NRTT*) with a standard deviation of 12ms (vs. 500ms). There were 13 desired data samples in each queue which contained between 4 et 16 samples.

At short distance, mean *GRTT* grows up to 1.2s (vs 23ms for *NRTT*) and the standard deviations are identical. There were 2 desired samples in each queue which size varied between 1 and 3 samples.

We observe the following relation ($s(t)$ is the evolution of the number of samples in each *NDR*) :

$$FRTT = 2 \times \max(s(t)) + NRTT$$

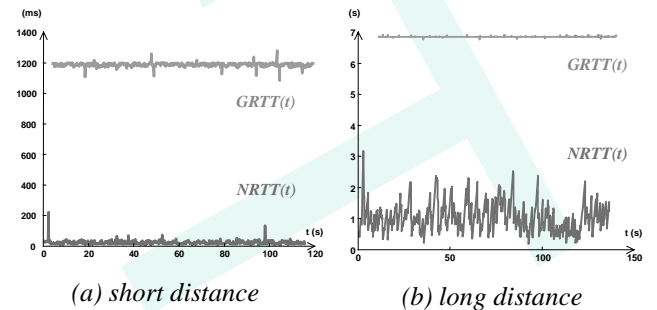


Figure 5 - Comparison between NRTT and GRTT

To put it in a nutshell, long distance results could be better with a real-time environment for *NDRs* which we didn't have at first for compatibility reasons. Nevertheless they inform us that the principle seems correct even if parameters might be better chosen to limit the embarrassing raising of the mean *GRTT*.

At short distance, mean RTT is multiplied by 50 but it is difficult to have it smaller because the queue sizes vary between 1 and 3 samples. In fact, it is due to the sampling period T_i of 200ms which is 10 times as much as the mean *NRTT*. As the mean *GRTT* is a multiple of T_i , the only mean to decrease mean *GRTT* is to decrease T_i .

4.2.4 Enhancement : dynamic adaptation

4.2.4.1 Principle

As the latest conclusions let us foresee, we need to adapt the transmission period T_i and the nominal size (*ns*) of the *NDR* queues. Consequently it is necessary to predict the evolution of network behavior in order to change the transmission parameters whenever they become out of proportions.

A change of these parameters leads the system to pause the teleoperation task and then to go through a new transient pattern as at the first starting. This involves that every calculus made from these parameters (state and network prediction, control laws, ...) are able to adapt themselves to the new parameters.

Network behavior prediction is achieved this way: we use two discrete *Kalman* filters coupled with a triple integrator model and an autocorrelation $\phi[k] = \sigma_Q^2 \cdot e^{-\alpha \cdot |k| \cdot T_i}$ [12]. This kind of association is typically used to predict movements of targets in problems of tracking. The first filter is used to predict the mean *NRTT* and the second one, the evolution of *mt*, the maximum time between two receptions (directly used for resizing the *NDR* queues). Kalman matrices are then :

$$A = \begin{bmatrix} 1 & T_i & \frac{T_i^2}{2} \\ 0 & 1 & T_i \\ 0 & 0 & 1 \end{bmatrix} \quad B = [0] \quad H = [1 \ 0 \ 0] \quad Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_Q^2 \end{bmatrix}$$

$$P_0 = I_3 \cdot \sigma_Q^2 \quad R = \sigma_Q^2$$

This filter gives us a prediction one step ahead, which corresponds to a few 100ms according to the transmission period. As we need to predict the network behavior farther ahead (a few seconds would be reasonable), we compute the prediction step of the *Kalman* filter 50 times: $\hat{x}_{k+I} = A^I \cdot \hat{x}_k$. As this extrapolation is noisy, we filter it with a low-pass filter ($\tau=50 \times T_i^*$). This gives us an extrapolation which we have found good enough to be used for further parameter adaptation.

The difficulties reside in measuring in one hand the mean *NRTT* when its standard deviation is high as in *TCP* transmissions, and in the other hand, the maximum time

between two successive frames : *mt(t)*.

To measure the mean *NRTT*, worked out from the elapsed time between the emission of a frame and the reception of its corresponding acknowledgement, we simply use a low-pass filter with $\tau=250 \times T_i$. This is a calculus we have found appropriate according to simulations we had done.

To measure the evolution of *mt(t)*, we compute the time between two successive frames *tsf(t)* and we keep the superior envelope of this signal. This computation consists in the following algorithm:

```

If (tsf[k] > e[k-1])
Then e[k] = tsf[k]
Else e[k] = e[k-1] / vd

```

Decreasing speed *vd* has been set to 1,00005. As this envelope contains high frequencies, we filter it with a first order discrete low-pass filter with $\tau=25 \times T_i$.

4.2.4.2 Validation by simulation

In order to validate our network behavior predictor, we have generated delays which mean value and standard deviation vary (see figure 6). We have chosen continuous sine variations with a discontinuity in order to represent long time variations of a real network. As we use a protocol that ensures data will be received in the same order as it has been sent, delays are modified to respect the right order of data reception. The prediction is tested during 20mn.

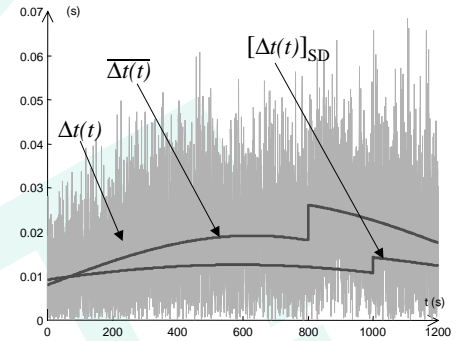


Figure 6 – Generated delays with mean and std. dev.

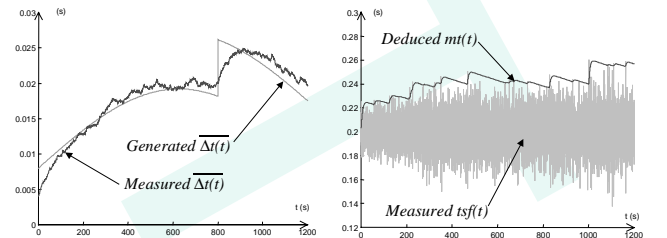


Figure 7 – Real and measured $\Delta t(t)$

The first *Kalman* filter gives a prediction of the mean delays (ie *NRTT*) with an advance of $50 \times T_i = 10s$. Figure 8 pictures the results. We can remark that the precision of the prediction depends a lot on the initial measure. In this case, the mean delays measure is a bit too slow when discontinuities happen. This prediction is then used to compute a new transmission period T_i at next parameter change.

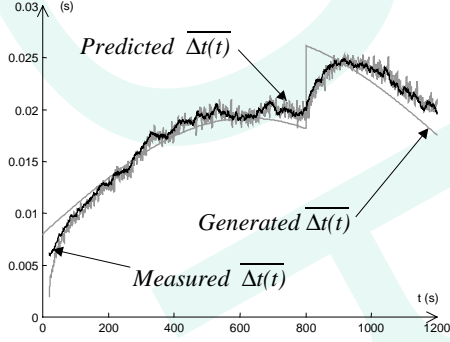


Figure 8 – Mean delays prediction

The second Kalman filter gives a 10s prediction of $mt(t)$, in other words, $mt(t+10s)$. The results showed in figure 9 are good enough to be used to prevent *NDR* queue emptying 10s before it would happen. It leaves enough time for the system to get in pause mode.

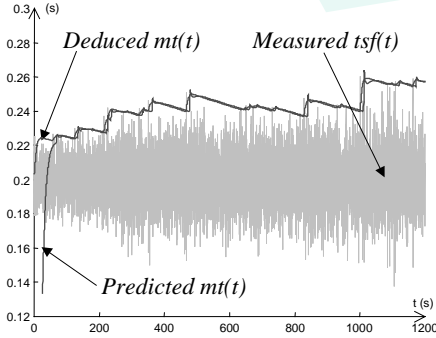


Figure 9 – Prediction of $mt(t)$

4.2.4.3 Computation of new parameters

Experimentations at short distance showed that the average *GRTT* was equal to 1.2s versus 23ms for *NRTT*. As $T_i=200ms$, the time spent in each *NDR* was equal to $3 \times T_i$. Therefore we had $GRTT=2 \times 3 \times T_i + NRTT$. In order to have a good resolution during the regulation, T_i should be at most equal to *NRTT*, but transmission frequency is limited by the network bandwidth (which may vary) and by the speed of computer hosted in the local and remote site. The choice of T_i will then depend on the platform used for the teleoperation. In our case, we have decided to make the following calculus:

$$T_i = \max(50ms, NRTT/3)$$

For our short distance, we would obtain an average *GRTT* equal to $2 \times 3 \times 50ms = 300ms$ versus 1,2s in first experiments. And at long distance $2 \times 16 \times 610/3ms \approx 3.9s$ (T_i would be equal to 200ms so the queues would have the same sizes as in our experimentations) versus 7s.

Concerning the nominal size of the *NDR* queues, we keep the same calculus for ns as in paragraph 4.2.1.

4.3 Estimation and prediction

We still use the generic estimation/prediction method presented in [10] and visible in figure 10. The principle consists in identifying the remote system behavior $H(z^{-1})$ and to use this identification to output an estimation of current state $x_{3e}[k]$ and a prediction of future state $x_{3p}[k]$ at the moment when orders $x_1[k]$ are emitted.

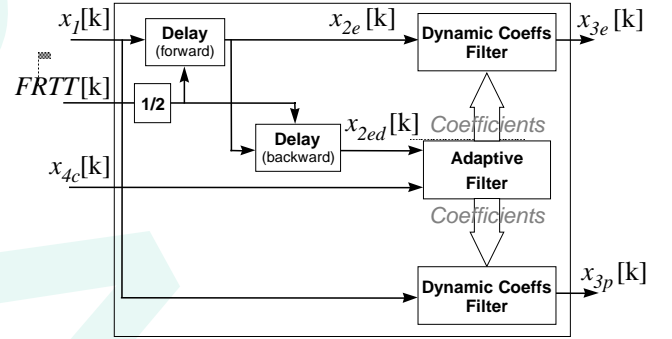


Figure 10 – Generic prediction/estimation diagram

In previous papers, we used a gradient adaptive filter. This filter has the advantages to be simple to use but it is not very stable and it is slow. Consequently, we have exchanged this filter with an identification method based on recursive mean squares algorithm. As every axis of the system can be modeled as a low-pass second order system (thanks to computed torque method [8]) and by way of bilinear transform, we identify every axis by the following discrete transfer function:

$$H_i(z) = \frac{b_{0i} + b_{1i} \cdot z^{-1} + b_{2i} \cdot z^{-2}}{1 + a_{1i} \cdot z^{-1} + a_{2i} \cdot z^{-2}}$$

Results of this part will make the object of future publication.

4.4 Experiments to be Held

We still have to finalize the experimental validation of our dynamic network delay regulator in miscellaneous teleoperation situations. We also plan to make experiments with the global mobile manipulator in a scenario described in [1] where only parts of experimentations have been carried out at this time.

5. CONCLUSION

The teleoperation project we have started for a few years begins to really take shape. Latest results depicted in this paper concern encouraging simulation results of network delay dynamic regulation. Estimation/prediction is presented here but results will be published later. We are working on experimentations to finally validate the whole system as a low level layer for teleoperation controls.

Up to now, this entire study was based on *TCP* protocol over Ethernet (LAN) and Internet (WAN) asynchronous networks. An application of this work will consist in incorporating every part of this work in a protocol layer over *IP*. This layer would provide specific quality of service and performances necessary to implement real-time control systems requiring transmissions, such as teleoperation.

6. REFERENCES

- [1] A. LELEVÉ, P. FRAISSE, P. DAUCHEZ & F. PIERROT, "*An enhanced mobile manipulator*", Proc. of the 4th World Automation Congress (WAC 2K), Maui, Hawaii, USA, 2000.
- [2] R. OBOE, P. FIORINI, "Internet-based telerobotics: problems and approaches", Proc Intl. Conf. on Advanced Robotics (ICAR'97), Monterey (CA - USA), July 1997, pp. 765—770.
- [3] G. NIEMEYER & J.-J. E. SLOTTINE, "*Designing Force Reflecting Teleoperators with Large Time Delays to Appear as Virtual Tools*", Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'97), pp. 2212—2218.
- [4] P. G. BACKES, K. S. TUO, J. S. NORRIS, G. K. THARP, J. T. SLOTTAD, R. G. BONITZ & K. S. ALI, "*Internet-Based Operations for the Mars Polar Lander Mission*", Proc. of the IEEE/ASME Intl. Conf. on Advanced Intelligent Mechatronics, September 1999, pp. 317—322.
- [5] A. LELEVÉ, P. FRAISSE, A. CROSNIER, P. DAUCHEZ & F. PIERROT, "*Towards Virtual Control of Mobile Manipulators*", Proc. of the 3rd World Automation Congress (WAC'98), Anchorage, USA, 1998.
- [6] C. PERRIER, P. DAUCHEZ, F. PIERROT, "A global Approach for Motion Generation of Non-Holonomic Mobile Manipulators", Proc. Of the IEEE Intl. Conf. On Robotics and Automation (ICRA'98), Leuven, Belgium, 1998, pp. 2971—2976.
- [7] E. FREUND, "*Fast nonlinear control with arbitrary pole placement for industrial robots and manipulators*", The Intl. Journal of Robotics Research, Vol. 1(1), pp. 65—78, 1982.
- [8] A. LELEVÉ, P. FRAISSE, P. DAUCHEZ & F. PIERROT, "*Teleoperation through the Internet: experimental results with a complex manipulator*", Proc. of the Intl. Symp. on Robotics (ISR'99), Tokyo, Japan, 1999.
- [9] A. LELEVÉ, P. FRAISSE, P. DAUCHEZ & F. PIERROT, "*Modeling and Simulation of Robotic Tasks Teleoperated through the Internet*", Proc of the Intl. Conf. on Advanced Intelligent Mechatronics (AIM'99), Atlanta, USA, 1999.
- [10] R. OBOE et P. FIORINI, "A Design and Control Environment for Internet-based Telerobotics", International Journal of Robotics Research, Vol. 17 N°4 April 1998, pp. 433—449.
- [11] S. MOON and P. SKELLEY and D. TOWSLEY, "*Estimation and Removal of Clock Skew from Network Delay Measurements*", Proc. of IEEE INFOCOM '99, New-York, NY, March 1999.
- [12] F. JANABI-SHARIFI, V. HAYWARD, C-S. J. CHEN, "*Novel Adaptive Discrete-Time Velocity Estimation Techniques and Control Enhancement Of Haptic Interfaces*", <http://citeseer.nj.nec.com/22148.html>.

* Every numerical value has been chosen considering several simulation cases.